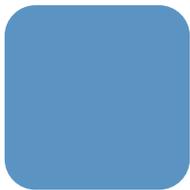
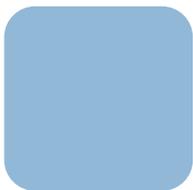


Steps required for creating a progress bar in Linux

Version 1.0
11.07.2019



Loading 50%.....



**Elektronik
Systeme**

© F&S Elektronik Systeme GmbH

Untere Waldplätze 23

D-70569 Stuttgart

Fon: +49(0)711-123722-0

Fax: +49(0)711 – 123722-99



About this document

This document tells us how to create a new Progress bar using a program in Linux. The document also covers some essential information from the parameters usage to the .ini file.

Table of Contents

About this document	2
Table of Contents	2
1 Progress bar	3
2 Fbsplash function	3



1 Progress bar

A Progress bar is a graphical control element used to visualize the progression of an extended computer operation, such as a download, file transfer, or booting. Sometimes, the graphic is accompanied by a textual representation of the progress in a percent format. So, everywhere we can find the wise implementation of progress bar.

There are certain steps which are needed to follow in order to make or design a new progress bar. In the next capital, it is described in detail on how to create a new progress bar in Linux while booting up the system.

2 Fbsplash program

Fbsplash is a userspace implementation of a splash screen for Linux systems. In simple words, it provides a graphical environment (e.g. Progress bar, background picture) during system boot/reboot using the Linux framebuffer layer.

When **fbsplash** is simply typed in a terminal program, then the usage of fbsplash program is displayed on the screen as shown in the figure 1. The usage of fbsplash program looks like this.

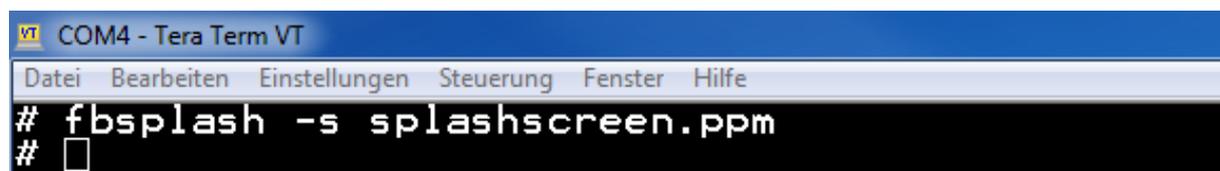
```
Usage: fbsplash -s IMGFILE [-c] [-d DEV] [-i INIFILE] [-f CMD]
-s      Image
-c      Hide cursor
-d      Framebuffer device (default /dev/fb0)
-i      Config file (var=value):
        BAR_LEFT, BAR_TOP, BAR_WIDTH, BAR_HEIGHT
        BAR_R, BAR_G, BAR_B, IMG_LEFT, IMG_TOP
-f      Control pipe (else exit after drawing image)
        commands: 'NN' (% for progress bar) or 'exit'
```

Figure 1: Usage of fbsplash program

*Note: Make sure not to forget **space** in between the parameters when typing the command in the terminal.*

As we see, there are mainly four parameters (**IMGFILE**, **DEV**, **INIFILE**, **CMD**) which needs to be configured, before giving “fbsplash” command to the terminal program.

IMGFILE is an image file based on **PPM** (Portable Pixmap) format. If the image is in different format **PNG/BMP**, then it should be converted to **PPM** format. Other forms of the image will not accept by the program fbsplash. The command **[-s IMGFILE]** executes the background image on the screen while booting up the system(see figure 2).



```
COM4 - Tera Term VT
Datei Bearbeiten Einstellungen Steuerung Fenster Hilfe
# fbsplash -s splashscreen.ppm
#
```

Figure 2: Command to displaying an Image

DEV is a framebuffer device and remains **/dev/fb0** as a default parameter. Here, nothing needs to be changed and the command is **[-d /dev/fb0]**.

```
COM4 - Tera Term VT
Datei Bearbeiten Einstellungen Steuerung Fenster Hilfe
# fbsplash -s splashscreen.ppm -d /dev/fb0
# █
```

Figure 3: Framebuffer device as a default parameter

INIFILE is an important parameter for designing and displaying the progress bar back on the screen. INIFILE is non-other than a configuration file, where several parameters for designing the progress bar have to be parameterized according to the user demand. INIFILE can be made simply on a notepad as shown in the figure 4.

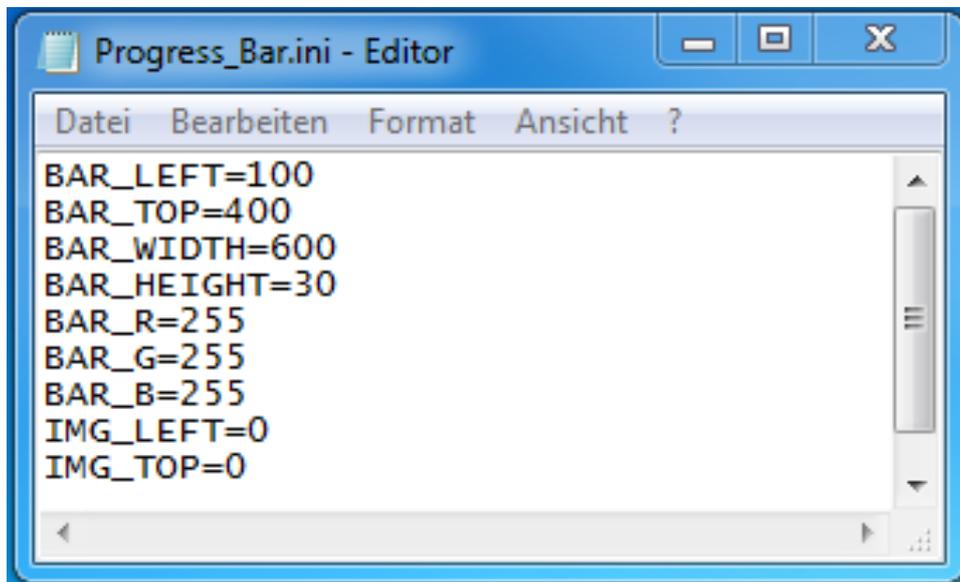


Figure 4: INIFILE containing parameters for designing the progress bar

Note: After downloading INIFILE in Linux, please check the file again because the file might contain unusual characters along with the values and they have to be removed from the file (see figure 5).

We can see nine different parameters therein *Figure 4*. Each different parameter plays different roles ranging from filling the background with colours to defining dimension and position of progress bar along with position of background image.

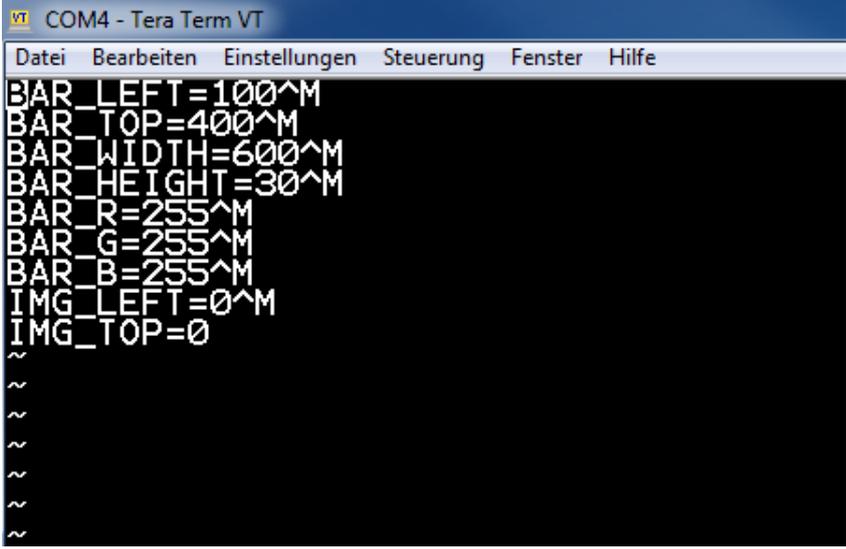
The values assigned to **BAR_LEFT** and **BAR_TOP** decides the position of progress bar, in which varying the value of **BAR_LEFT** shifts the progress bar towards the left/right on the screen i.e. horizontal movement, whereas the progress bar does the vertical movement using **BAR_TOP**.

Similarly, **BAR_WIDTH** and **BAR_HEIGHT** define dimension of progress bar. As name **BAR_WIDTH** changes width of bar and **BAR_HEIGHT** changes height of bar

BAR_R, **BAR_G** and **BAR_B** are **RGB** (Red Green Blue) values and provide background colour to progress bar. Changing the values of these parameters change the background colour of progress bar. Unfortunately, there is no any other parameter that can change

foreground colour of progress bar. The colour of foreground is grey and is constant throughout whole time.

The last two parameters **IMG_LEFT** and **IMG_TOP** moves background image towards right and down the screen respectively. Basically, it's not necessary to bring change in their values and the ideal value for both of them remains **0**.

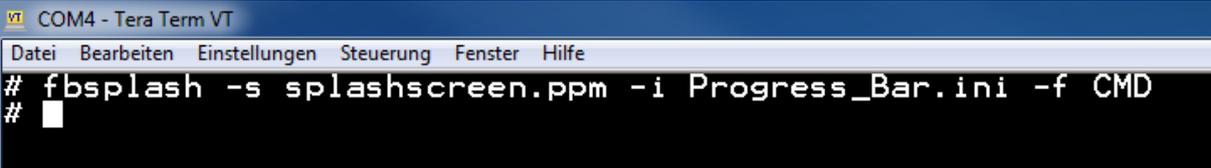


```
COM4 - Tera Term VT
Datei Bearbeiten Einstellungen Steuerung Fenster Hilfe
BAR_LEFT=100^M
BAR_TOP=400^M
BAR_WIDTH=600^M
BAR_HEIGHT=30^M
BAR_R=255^M
BAR_G=255^M
BAR_B=255^M
IMG_LEFT=0^M
IMG_TOP=0
~
~
~
~
~
~
~
```

Figure 5: Unusual characters ^M appears inside INIFILE

Note: The command **-I INIFILE** only works, when **-f CMD** is executed together (see figure 6). Otherwise progress bar won't appear on the screen. Don't execute the command now!

The successful configuration of INIFILE brings us to the next parameter **[-i CMD]** in fbsplash program. This command allows us to work with percentage coverage from 0 to 100 in progress bar.



```
COM4 - Tera Term VT
Datei Bearbeiten Einstellungen Steuerung Fenster Hilfe
# fbsplash -s splashscreen.ppm -i Progress_Bar.ini -f CMD
# █
```

Figure 6: Command to percentage coverage in progress bar

For example, Figure 7 shows the 40% coverage of progress bar after the execution of the command.

The last work is just to add the whole command line (**fbsplash -s IMGFILE -I INIFILE -f CMD**) to **/etc/init.d/** directory with different percentage coverage (0%, 20%, 40%, 60%, 80%, 100%) and execute it. **Everything must work now !**

Note: It seems to appear still a bit of space in progress bar even though the percentage coverage is 100 %.



Figure 7: 40% coverage of progress bar

Figure

- Figure 2: Usage of fb splash function.....3*
- Figure 2: Command to displaying an Image.....3*
- Figure 3: Framebuffer device as a default parameter.....4*
- Figure 4: INIFILE containing parameters for designing the progress bar.....4*
- Figure 5: Unusual characters ^M appears inside INIFILE.....5*
- Figure 6: Command to percentage coverage in progress bar.....6*
- Figure 7: 40% coverage of progress bar.....7*

